1. A method of looking up a key associated with a packet to determine a route through a routing device comprising:

upon receipt of the key, forward traversing one or more nodes which make up a trie stored in a memory by evaluating at each node traversed one or more bits in the key as indicated by a bits-to-test indicator associated with each node, a value of the bits in the key determining a path traversed along the trie;

locating an end node in the trie, the end node having a route;

comparing the route to the key;

if they match, outputting destination information associated with the end node to guide the transfer of the packet through the routing device; and

if they do not match, traversing the trie backwards to locate a best match for the key.

2. The method of claim 1 wherein the step of forward traversing includes storing on a stack, for each node having one or more attached routes, the bits-to-test indicator and pointers to the attached routes.

3. The method of claim 2 wherein the step of traversing the trie backwards includes:

comparing the key with the route to determine a first dissimilar bit location;

popping entries off the stack to determine when the bits-to-test indicator associated with a first node in the backward traversal is less than or equal to the first dissimilar bit location; and

outputting destination information associated with the first node to guide the transfer of the packet through the routing device.

4. The method of claim 3 wherein the first node includes a route, the method further including calculating statistical information for each packet and storing in memory the statistical information with the route associated with each of the end node and the first node.

5. The method of claim 4 further including transferring the statistical information along with the destination information to an output port in the routing device for transfer to a destination.

6. The method of claim 1 further including the step of prior to a forward traversal of the trie, searching a root table for a match of a predetermined number of bits in the key, the root table indexed by the predetermined number of bits where each entry includes a pointer to a start node in the trie to begin the forward traversal.

7. The method of claim 1 wherein the forward traversal of the trie includes loading node information for each node traversed in the trie until the end node is reached, the node information including a bits-to-test indicator, a plurality of child pointers and one or more attached routes.

8. The method of claim 1 where the nodes in the tree include 2^n child pointers, where N is an integer greater than 1, each pointer including a bits-to-test indicator and an address in the memory where the child node is stored, the bits-to-test indicator indicating one or more bits to be tested in the key associated with a child node to which the child pointers indicate, and where the forward traversal includes testing one or more bits indicated by the bits-to-test indicator and retrieving an appropriate child pointer associated with the child node that indicates a next node in the trie to be traversed after the child node, whereby accesses to the memory are minimized in the forward traversal of the trie by loading a single pointer at each node until the end node is reached.

9. A method of routing a packet through a switch comprising:

upon receipt of the packet, extracting a key from the packet;

forward traversing a trie by evaluating at each node one or more bits in the key as indicated by a bits-to-test indicator associated with each node, values of the bits in the key located at a position indicated by the bits-to-test indicator determining a path traversed along the trie at each node;

locating an end node in the tree, the end node having a route;

comparing the route to the key;

if they match, retrieving destination information associated with the end node;

if they do not match, traversing the trie backwards to locate a best match for the key having a route and destination information associated therewith; and

routing the packet through the switch according to the destination information.

10. A method of inserting a route in a route table where the route table is stored as a trie in a memory of a routing device, the route table defining a path by which a packet is transferred through the routing device, the method comprising:

traversing the trie to determine an insertion point;

determining if the insertion point has an associated parent node and sibling node in the trie;

if so, creating a multi-node from the parent, sibling and the route including setting one or more child pointers in the multi-node to indicate a node directly beneath the insertion point;

storing the multi-node in the memory; and

updating the child pointer in a node directly above the parent node to indicate a starting address in the memory for the multi-node.

11. A router for routing packets in a packet switched network comprising:

one or more input ports for receiving packets;

a packet memory;

an input switch coupled to each input port and the memory, the input switch including a transfer engine for transferring packets from an input port to the packet memory and a key extraction engine for extracting a key from each packet;

a controller coupled to the input switch, the controller including a key look-up engine and a route memory, the route memory for storing a route table where the route table includes a trie, the key look-up engine traversing the trie to determine a best match to the key, and upon determining the best match for the key, generating notification information, the key look-up engine including a forward traversal engine for forward traversing the trie operable to evaluate at each node traversed one or more bits in the key as indicated by a bits-to-test indicator associated with each node, where the values of the bits in the key determine the path traversed along the trie, locate an end node having a route, compare the route to the key, if they match, outputting destination information associated with the end node to guide transfer of a packet through the routing device, and if they do not match, traversing the trie backwards to locate a best match for the key;

one or more output ports;

an output switch coupled to the controller, the packet memory and the output port for transferring packets from packet memory to an appropriate output port based on the notification information received from the controller.

12. The apparatus of claim 11 further including a stack and where the forward traversal engine stores on the stack a bits-to-test indicator for the node and pointers to attached routes for each node having attached routes.

13. The apparatus of claim 12 where the key look-up engine includes a backward traversal engine operable to compare the key with the route to determine a first

dissimilar bit location, pop entries off the stack to determine when a bits-to-test indicator associated with a first node in the backward traversal is less than or equal to the first dissimilar bit location, and output destination information associated with the first node to guide a transfer of a packet through the router.

14. The apparatus of claim 11 further including a statistical engine for calculating statistical information for each data packet and outputting the statistical information with destination information for transfer to a destination port.

15. The apparatus of claim 11 where the route memory is divided into a plurality of banks, and where parent and children nodes in the tree are stored in different banks.

16. The apparatus of claim 11 where each node in the tree includes a bits-to-test indicator, 2^n child pointers, and n or fewer attached routes, where a pair of the 2^n child pointers indicate a child node to be traversed in the forward traversal when a value of a bit in the key as indicated by the bits-to-test indicator is a first value and second value, respectively.

17. The apparatus of claim 16 where the child pointers and attached routes are stored in contiguous locations in the route memory.

18. The apparatus of claim 11 wherein the trie is a modified radix trie.

19. The method of claim 18 where the nodes in the modified radix trie include 2^n child pointers, each pointer including a bits-to-test indicator and an address in the memory where the child node is stored, the bits-to-test indicator indicating one or more bits to be tested in the key associated with a child node to which the child pointers

indicate, and where the forward traversal engine is operable to test one or more bits indicated by the bits-to-test indicator and retrieve an appropriate child pointer associated with the child node that indicates a next node in the trie to be traversed after the child node, whereby accesses to the memory are minimized in the forward traversal of the modified radix trie by loading a single pointer at each node until the end node is reached.

20. A route look-up engine for locating a best match for a key in a route table, the route table including a trie stored in a memory associated with a routing device, the trie including one or more entries defining a path through a routing device for transferring a packet in a packet switched network from a source to a destination, the route look-up engine comprising:

a stack for storing stack entries including a bits-to-test indicator and a pointer to the destination; and

a plurality of look-up engines each including a buffer, a bit comparison engine and a key comparison engine, the buffer for storing node information that is retrieved from the memory, the node information including a bits-to-test indicator,

wherein responsive to receiving the key associated with a packet, the look-up engine forward traverses the trie, the bit comparison engine evaluating at each node traversed one or more bits in the key as indicated by a bits-to-test indicator associated with each node, a value of the bits in the key determining the path traversed along the trie, the look-up engine storing stack entries on the stack for each node traversed having an attached route, the look-up engine locating an end node having a route, the key comparison engine performing a singular key comparison for each packet routed through the routing device by comparing the key with the route, if they match, the look-up engine outputting destination information associated with the end node to guide the transfer of the packet through the routing device and if they do not match, the look-up engine traversing the trie backwards, popping entries off the stack, to locate a best match for the key and destination information associated therewith.

21. A computer program, tangibly stored on a computer-readable medium, comprising instructions for causing a computer to:

upon receipt of a key, forward traverse a trie stored in a memory by evaluating at each node traversed one or more bits in the key as indicated by a bits-to-test indicator associated with each node, a value of the bits in the key determining a path traversed along the tree;

locate an end node in the tree, the end node having a route;

compare the route to the key;

if they match, output destination information associated with the end node to guide a transfer of a packet through a routing device; and

if they do not match, traverse the trie backwards to locate a best match for the key and destination information associated therewith.

22. The computer program of claim 21 wherein the forward traverse includes instructions to cause the computer to:

store on a stack for each node having an attached route the bits-to-test indicator for the node and pointers to any attached routes.

23. The computer program of claim 22 wherein the backward traverse includes instructions to cause the computer to:

compare the key with the route to determine a first dissimilar bit location;

pop entries off the stack to determine when the bits-to-test indicator associated with a first node in the backward traversal is less than or equal to the first dissimilar bit location; and

output destination information associated with the first node to guide a transfer of the packet through the routing device.

24. A router for switching a data packet between a source and destination in a network comprising:

an input port including a data handler, the input port receiving the data packet from the source, the data handler dividing the data packet into one or more fixed length cells;

an output port for routing the data packet to the destination;

a memory divided into a plurality of memory banks;

an input switch for receiving the fixed length cells from the input port and routing consecutive cells of the data packet to different memory banks, wherein a single cell is transferred in a cell slot time span to a memory bank, the input switch including a key reading engine for extracting key information from a first cell received at the input switch associated with the data packet;

a controller coupled to the input switch and receiving the key information therefrom, the controller for decoding destination information from the key information received from the input switch and outputting a notification defining a routing of the data packet from the memory to the output port, the controller including a plurality of look-up engines and a memory for storing routes, each look-up engine operable to compare the key information with routes stored in the memory and determine a best route through the router for a given data packet; and

an output switch for routing cells received from the memory to the output port.

25. The router of claim 24 where the input switch includes a linking engine for linking cells in the data packet to allow retrieval of the data packet from non-contiguous locations in the memory.

26. The router of claim 25 further including an indirect cell generator for generating one or more indirect cells, the linking engine tracking the location in the memory where consecutive cells of the data packet are stored and providing an address in memory of each cell in the data packet for storage in indirect cells.

27. The router of claim 24 wherein the input switch time division multiplexes the writing of data packets to the memory such that consecutive cells from the input port are written to consecutive banks in the memory.

28. The router of claim 27 wherein the output port includes a result processor for receiving the notification from the controller and initiating a transfer of the data packet from the memory to the output port.

29. The router of claim 24 wherein the input switch includes a reservation table for scheduling transfers from the memory to the output switch.

30. The router of claim 29 wherein the output switch routes the notification to the output port and thereafter the output port issues a request to the input switch to transfer the data packet from memory to the output port through the output switch.

31. The router of claim 30 wherein the request from the output port is stored in the reservation table.

32. The router of claim 31 wherein requests to transfer cells from memory to the output switch are time domain multiplexed so that during one cell slot time span at most a single read request is issued to each bank in the memory for servicing.

33. The router of claim 30 wherein the memory outputs at most a single cell per bank in one cell slot time span.

34. A method implemented in a router for switching a data packet between a source and destination in a network, the data packet including a header portion and a data portion, the header portion including routing information for the data packet, the method

comprising:

defining a data path in the router comprising a path through the router along which the data portion of the data packet travels;

defining a control path comprising a path through the router along which routing information from the header portion travels;

separating the data path and control path in the router such that the routing information can be separated from the data portion allowing for the separate processing of each in the router; and

storing the data portion in a global memory while routing decisions are made on the routing information in the control path.